Attorney Docket No. 76580.P012 *PATENT*

UNITED STATES PATENT APPLICATION

FOR

LINEAR SPECTRAL FREQUENCIES CODING FOR LOW BIT RATE SPEECH COMPRESSION

INVENTORS:

Victor D. Kolesnik
Boris D. Kudryashov
Eugeny Ovsjannikov
Sergey Petrov
Boris Trojanovsky

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026
(408) 720-8598

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EL546447178US

Date of Deposit August 25, 2000

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231

Sheena Hicks
(Typed or printed name of person mailing paper or fee)

Sheena Hicks 8-25-2000
(Signature of person mailing paper or fee) Date

LINEAR SPECTRAL FREQUENCIES CODING FOR LOW BIT RATE SPEECH COMPRESSION

NOTICE OF RELATED APPLICATION

This is a continuation of US Provisional Patent Application Number 60/157,647, entitled "Method And Apparatus For A Linear Spectral Frequency Audio Compression," filed 10/4/99.

FIELD OF INVENTION

The invention relates to low rate speech coding in communication and data processing systems, and more particularly to spectrum quantization of voice signals

BACKGROUND OF THE INVENTION

Digital speech processing is extensively used in communication systems, telephony, digital answering machines, low rate videoconferencing, etc. Low rate speech coding is typically based on parametric modeling of the speech signal. The speech encoder computes representative parameters of the speech signal, quantizes them into products, and places them into the data stream, which may be sent over a digital communication link or saved in a digital storage media. A decoder uses those speech parameters to produce the synthesized speech.

Almost all known speech compression algorithms for bit rates less than or equal to 8000 are based on linear prediction. Typically, linear prediction coefficients (LPC) are transmitted as linear spectral frequencies (LSF) (sometimes they are called "linear spectral parameters (LSP)" or "linear spectral pairs (LSP) "). Depending on the bit rate provided by the speech compression algorithm, LSF are updated once per 10-30 ms. Usually a 10th order linear prediction filter is used, which means that the LSF are represented by a 10-dimensional vector.

Figure 1 is a block diagram of a typical LSF encoder based on vector quantization. The current frame of a digitized speech signal enters the LSF calculator unit 110 where the current LSF vector is computed. Previous quantized LSF vectors are kept in the buffer memory 150. Typically only one last previous vector is stored in the buffer memory 150. The LSF predictor unit 160 computes some predetermined number of LSF vector predicted values. Indeed, some of these predicted values are typically independent of previous LSF vectors.

Then the current LSF vector and the set of predicted LSF vectors enters the vector quantizer unit 120. The vector quantizer unit 120 determines the best codebook index (or set of indices) and the best predictor number to provide the best approximation of the current LSF vector in the sense of some distortion measure. All indices computed by the vector quantizer enter indices encoder

unit 130 where they are transformed into the codeword corresponding to the current LSF vector.

This codeword is sent along with other speech parameters into a data link transmission medium or a digital memory. Also, the codebook indices and predictor index enter the LSF reconstruction unit 140. Another input of the reconstruction unit is the set of predicted LSF vectors. In the LSF reconstruction unit 140 the quantized LSF vector is reconstructed. This vector is then saved in the buffer unit 150 to be used for prediction next LSF vectors.

Early quantizers used a single non-structured code and compared the source vector to each entry in the codebook (referred to as "full search quantizers"). The performance of vector quantization depends on the size of the codebook used, and to obtain better results, larger codebooks have to be used. On the other hand, storage and processing complexities also increase with increasing codebook size. To overcome this problem, suboptimal vector quantization procedures have been proposed that use multiple structured codebooks. One of the most widely used procedures is multistage vector quantization (MSVQ). In MSVQ a sequence of vector quantizers (VQ) is used. The input of the next VQ is the quantization error vector of the previous VQ.

An improvement on MSVQ is M-best or delayed decision MSVQ, which is described in (W.P. LeBlanc, B. Bhatacharya, S.A. Mahmood and V. Cuperman, "Efficient search and design procedures for robust multistage VQ of LPC Parameters for 4 kb/s speech coding" *IEEE Transactions on speech and audio*

processing. Vol. 1, No. 4, October 1993, pp. 373-385). The M-best MSVQ achieves better quantization results by keeping from stage to stage a few candidates (M candidates). The final decision for each stage is made only when the last quantization stage is performed. The more candidates that are kept, the higher the quantization that may be achieved and the greater the computational complexity.

The unit having the greatest impact on the performance of the quantizer is the vector quantization unit. Typically, an LSF vector is split into subvectors (usually 1 to 3 subvectors). A vector quantization procedure is then applied to each subvector. To improve the quantization accuracy, it is necessary to increase the dimensions of the subvectors and the corresponding codebook sizes. However, this leads to increasing the computational load needed for full search quantization. To decrease computational complexity, a multistage M-best quantization procedure is used.

The block diagram of a two-stage M-best quantizer is shown in **Figure 2**. A source vector enters the first quantizer 210 having a smaller structured codebook C_1 of size L_1. For each entry x of the set of L_1 codewords, the residual, or error vector is computed by subtracting x from the source vector. The output of this quantizer is a set of M_1 codewords closest to the source vector in the sense of some distortion measure. The error vectors are processed by the second quantizer 220 with a smaller structured codebook C_2 of size L_2. The resulting candidate code vector(s) are then obtained as component wise sums of

the first quantizer output and the corresponding approximated errors by adder 230. The final decision is made by the select best codeword unit 240 which selects from among the candidates the candidate closest to the source vector.

The common property of these suboptimal vector quantizers is that they reduce computational complexity by replacing an optimal large size non-structured codebook with a direct sum of small structured codebooks.

SUMMARY OF THE INVENTION

A reduced complexity vector quantizer is described. According to one embodiment of the invention, a multistage vector list quantizer comprises a first stage quantizer to select candidate first stage codewords from a plurality of first stage codewords, a reference table memory storing a set of second stage codewords for each first stage codeword, and a second stage codebook constructor to generate a reduced complexity second stage codebook that is the union of sets corresponding to the candidate first stage codewords selected by the first stage quantizer.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements, and in which:

Figure 1 (prior art) is a block diagram illustrating a general structure of an LSF encoder based on vector quantization.

Figure 2 (prior art) is a block diagram illustrating a general structure for two-stage M-best vector quantization.

Figure 3 is a block diagram of a two-stage list quantizer according to one embodiment of the invention.

Figure 4 is a block diagram illustrating a reduced complexity quantizer that uses a non-structured codebook according to one embodiment of the invention.

Figure 5A (prior art) illustrates the result of the combined first and second structured codebooks of a two-stage vector quantizer.

Figure 5B (prior art) shows the 4 codewords of the first stage codebook.

Figure 5C (prior art) shows the 4 codewords of the second stage codebook (see asterisks).

Figure 6 illustrates the design of a non-structured codebook of a two-stage list quantizer according to one embodiment of the invention.

Figure 7 is a block diagram of a general LSF encoder based on a multistage list quantizer (MSLQ) according to one embodiment of the invention.

Figure 8 illustrates the bit allocation of 16 bits per LSF MSLQ-based LSF quantizer according to one embodiment of the invention.

DETAILED DESCRIPTION

In the following description, numerous specific details are set forth to provide a thorough understanding of the invention. However, it is understood that the invention may be practiced without these specific details. In other instances, well-known structures and techniques have not been shown in detail in order not to obscure the invention.

Figure 4 is a block diagram illustrating a reduced complexity quantizer that uses a non-structured codebook according to one embodiment of the invention. **Figure 4** shows a searching unit 401 and a quantizer 405. The searching unit 401 includes a non-structured codebook C 402. Both the searching unit 401 and the quantizer 405 received the same source vector. The searching unit 401 uses a technique to dynamically select a subset of the codewords from the non-structured codebook C to form a reduced complexity codebook based on the current source vector. This reduced complexity codebook is provided to the quantizer 405.

The technique used by the searching unit 401 to select codewords from the non-structured code book C 402 to dynamically form the reduced complexity code book from the current input source vector depends on the implementation. However, the technique used will operate by performing less than a comparison of the source vector to every codeword in the codebook C. In particular, assume the codebook C includes L codewords. The searching unit will identify a subset of the L codewords without comparing the current source

vector to each of the L codewords. The reduced complexity codebook is then used by the quantizer 405 to quantize the source vector. As such, the source vector is quantized with a subset of the codewords from the original non-structured codebook C, rather than a direct sum of small structured codebooks as used in MSVQ techniques. In addition, the system of **Figure 4** uses a non-structured codebook, without performing all the comparisons required by the prior art full search quantizer. While various techniques can be used to implement the searching unit, several such embodiments are described herein with reference to Figures 3 and 5-7.

Figure 3 is a block diagram of a two-stage list quantizer according to one embodiment of the invention. The advantage of this quantizer over prior art suboptimal quantizers is that the computational complexity is reduced without a loss of quantization accuracy. Let C be a codebook with L k-dimensional vectors, generated, for example, by a well-known procedure, such as an LBG algorithm. The Multistage List Quantizer (MSLQ) 300 starts with a "coarse" pre-quantization of the source vector in the first-stage quantizer 310. First-stage quantizer 310 has a first stage codebook C_1 containing L_1 first stage codewords labeled x_1 to x_{L_1}. Its output is the first stage list of indices of M_1 codewords $\{j_1, \ldots, j_{M_1}\}$ closest to the source vector.

This list enters second-stage reduced complexity codebook constructor 330. The second-stage reduced complexity codebook constructor 330 is coupled to reference table memory unit 340. For each index of a codeword from first

stage codebook C_1, the reference table memory unit 340 keeps a precomputed set of P indices of second stage codewords from C. The second stage codebook C_2 is dynamically constructed by selecting codewords from C based on this table. In particular, let $C_2(j)$ denote the subset of C corresponding to the x_{ith} codeword from C_1. The second-stage reduced complexity codebook construction unit creates the second stage reduced complexity codebook as a union of the subsets $C_2(j)$, where $j = 1, \ldots, M_1$; reduced complexity codebook C_2 having L_2 codewords comprising = $\bigcup_{i=1}^{M_1} C_2(j_i)$.

The second stage reduced complexity codebook enters second-stage quantizer 320. The second-stage quantizer selects the best (closest to source vector) codeword from among the codewords of the reduced complexity codebook. This index of the codeword is the output of quantizer 300.

Thus, the searching unit of **Figure 3** uses a codebook C with L k -dimensional vectors, generated, for example, by a well known procedure, such as an LBG algorithm. The first-stage quantizer 310 uses a smaller codebook C_1 with L_1 codewords (where $L_1 < L$) based on C to quantize the source vector. The reduced codebook constructor 330 uses the codewords or indices of codewords selected by the first stage quantizer 310 to identify sets of P codewords, where $L/L_1 < P$, from the reference table 340. The reduced codebook constructor combines the identified sets to create the reduced codebook C_2 having L_2 codewords from C.

Figure 5A (prior art) illustrates the result of the combined first and second structured codebooks of a two-stage vector quantizer. The letters *a-d* symbolize codewords for the first stage codebook, and the numbers *0-15* symbolize codewords for the second stage codebooks. The first stage codewords are evenly distribured to cover the full spectrum of possible frequencies. The codewords for the second stage codebooks are evenly distributed to cover the areas represented by the first stage codewords. The codewords *0-3* cover the region of frequencies corresponding to codeword *a*. Figures 5B and C illustrate individually the structured codebooks of the first and second stage quantizers. Figure 5B (prior art) shows the 4 codewords of the first stage codebook. Figure 5C (prior art) shows the 4 codewords of the second stage codebook (see asterisks). Effectively, whatever codeword(s) O_x is selected from the first stage codebook, the second stage codebook $*_i$ through $*_{i+3}$ is applied to each selected O_x.

Figure 6 illustrates the design of a non-structured codebook of a two-stage list quantizer according to one embodiment of the invention. Consider the case when codebook sizes of both first and second quantizers are equal to 5. The 5 codewords of the first quantizer are labeled by the letters *a, b, c, d,* and *e*. The entire 16-word second stage codebook is partitioned into 5 intersecting subsets consisting of 5 points each, as shown in the **Figure 6**. In each subset, the 5 points closest to a codeword from first quantizer are included. This partitioning is kept in the reference table memory shown in

Figure 3 and **Figure 4**. For the example shown in **Figure 6**, this table may be shown in the form shown in Table 1. In Table 1, the codewords are enumerated as shown in **Figure 6**. The quantization method uses a first-stage codebook of size L_1, a second-stage codebook of size L_2, and a list size M as (L_1, L_2, M) -scheme. The considered example represents (5,5,1)-scheme. The MSE for this scheme and other list quantization schemes for rate 2 dimension 2 case are given in Table 2.

1st codebook word	Second codebook
A	0,3,5,7,8
B	1,2,4,6,8
C	5,7,10,11,13
D	8,11,12,13,14
E	6,9,12,14,15

TABLE 1

Quantization scheme (L_1, L_2, M)	MSE	Complexity
(5,5,1)	0.110	10
(5,6,1)	0.108	11
(5,7,1)	0.105	12
(5,5,2)	0.105	13.49

TABLE 2

MSE and complexity of some list quantization schemes for 16 codewords 2-dimensional quantizers

The complexity κ_2 of the multistage list quantizer shown in **Figure 3** is

$$\kappa_2 = L_1 + \left|\bigcup_{i=1}^{M} C_2(j_i)\right| \leq L_1 + ML_2, \qquad (2)$$

where L_1 and L_2 are the sizes of first-stage and second-stage codebooks, and M is the number of candidates kept after the first stage, and $C_2\,(j_i)$ denotes the

second-stage codebook corresponding to codeword j_i of the first-stage codebook. The total number of codewords is, in general, less than $L_1 L_2$. Note that the value of κ_2 depends on the list of candidates $(j_1,\ldots,j_M)$ chosen by the first-stage quantizer. It means that the complexity of this scheme is a random variable, but is upper bounded by the right side of inequality (2).

For example, consider a (5,5,2)-scheme. **Figure 6** and Table 1 show that depending on the 2 words chosen by the first-stage quantizer, the second-stage quantizer will search for the best codeword among 8 or 9 candidates. For instance, if first-stage quantizer chose pair *{a, b}* as a list, then the number of candidates is equal to 9, if the pair *{a,* c} is chosen, then number of candidates is equal to 8. Taking into account that first stage quantizer computes the error 5 times, the total complexity of (5,5,2)-scheme is estimated as 13.49.

Complexities of different 16-word 2-dimensional quantizers are given in Table 2. Note that (5,7,1) and (5,5,2) methods provide the same quantization quality as a prior art full search quantizer and requires fewer computations. At the same time conventional two-stage M-best quantizers can not provide this quality level irrespectively of the computational complexity. In general, the computational load may be reduced 4-5 times for 4-5 dimensional codebooks of size equal or greater than 512 codewords.

The MSLQ, in a two-stage embodiment, may use two codebooks: RQC (rough quantization codebook) and FQC (fine quantization codebook). Also, the MSLQ can store the reference table information describing each RQC entry, the

indices of some predetermined number FQC entries surrounding the RQC vector. MSLQ 300 can implement the following steps. Use an RQC for input vector quantization, and select a predetermined number of candidates. Then, construct a second-stage codebook. This subbook is union of FQC subsets corresponding to selected candidates in reference table. Among the second-stage codebook entries, choose the one closest to input vector in the sense of predetermined distortion measure. Use it's FQC index as a codeword.

This method may be used for more than two quantization stages. For this purpose the sequence of codebooks of increasing size have to be constructed. For each of the previous-stage codewords, the predetermined number indices of the next-stage codewords surrounding that previous-stage codeword are kept in the reference table. Quantization starts with list quantization using the smallest codebook. Then using reference table(s) the second stage codebook is constructed as a union of the sets corresponding to the candidates chosen on the first stage, etc. The final quantization result is one of largest codebook entries. Its index is a codeword corresponding to current LSF vector.

An alternative embodiment of vector quantization utilizing MSLQ shown in **Figure 3** is shown in **Figure 7**. A set of predicted LSF vectors (e.g., one or more vectors reconstructed from previous quantized LSF vectors) enter the first-stage quantizer unit 710 to be used as part of a codebook that includes a set of standard LSF vectors. In addition, the current LSF vector enters the first-stage quantizer unit 710. The first-stage quantizer 710 selects a

predetermined number of candidates from the codebook that provide the best approximation of current LSF vector in the sense of some distortion measure. The output of first-stage quantizer 710 is the list of indices of the chosen candidates with corresponding prediction error vectors. The list of indices and error vectors enter switch unit 720. The switch 720 forwards each error vector to either the first splitting means 730 or to the second splitting means 740 depending on the corresponding candidate index. For example, the error vector for the predicted LSF may be forwarded to first splitting means 730, and the error vector for the standard LSF vectors may be forwarded to second splitting means 735.

Further processing of error vectors is performed by two independent branches. These branches differ one from another in parameters of splitting means and codebooks used for subvectors quantization. It is clear that generally speaking any number of processing branches may be used in another embodiment of the present invention. Those vectors that enter first splitting means 530 are split into a predetermined number of subvectors of smaller dimension. In this embodiment the input vectors are split into 2 subvectors each. Then each subvector is quantized by a corresponding MSLQ unit 740, 750. A similar processing occurs in second splitting means 735 and MSLQ units 760 and 770. Each of the MSLQ units may have its own set of codebooks different from codebook used by other MSLQ units. The outputs of the MSLQ units are sets of quantized subvectors along with corresponding codebook indices. This

information enters the select best candidate unit 780, where a final decision about the best candidate is made. The output of quantizer contains the index of the best candidate and indices of 4 codebooks calculated in MSLQ units 740, 750, 760, 770.

The split-vector modification of the MSLQ of **Figure 3** used by the apparatus of **Figure 7** is referred to herein as split multistage list quantization (SMSLQ). In one embodiment, the SMSLQ-based method for quantizing a sequence of LSF vectors consists of the following steps: calculate an LSF vector for the current frame and calculate a set of predicted LSF vectors; calculate distance measure between the current LSF and codewords in a codebook including the set of predicted LSF vectors and a set of standard LSF vectors, select a predetermined number of candidates from the codebook having a minimal distortion measure; send the error vectors for the candidates for SMSLQ; and apply SMSLQ with different codebooks *C(j)* for quantizing the error vectors e(j), where *j* denotes the candidate index; select the one of the candidates for which that candidate and its quantized error vector provides the best approximation of the current LSF vector in the sense of a given distortion measure; and construct the fixed length codeword as a concatenation of a variable rate encoded candidate index and the variable rate encoded quantized error vector.

As indicated above, the codebook (or set of candidates) used by the first-stage quantizer 710 includes 2 parts: a standard part and an adaptively varying

part. The varying part is represented by the set of predicted LSF vectors. Variable length codewords are assigned to the candidates, because predicted LSF vectors usually are chosen more frequently than the standard LSF vectors. To satisfy this requirement, variable size codebooks are used for the second-stage (SMSLQ) quantization.

The advantage of MSLQ quantization over prior art MSVQ quantization is that MSLQ achieves the same quality as an exhaustive search over the FQC codebook, whereas the set of MSVQ-quantized vectors is direct sum of the stage codebook. The non-structured FQC codebook provides significantly better quantization accuracy than the structured codebooks used in the traditional multistage M-best quantization procedure.

The performance of this embodiment can be compared with the performance of other LSF coding schemes using a weighted Euclidean distance measure which is widely used in speech coding. This weighted distance (WD) $d(\mathrm{f,f'})$ between the input vector $\mathrm{f} = (f',\ldots,f_p)$ and the quantized vector $\mathrm{f'} = (f_1',\ldots, f_p')$ is given by

$$d(\mathbf{f},\mathbf{f'}) = \sum_{j=1}^{p} w_j (f_j - f_j')^2, \qquad (3)$$

where p is the number of elements in f , and w_j is a weight assigned to the j th frequency. p = 10 in this example. Also, weighting coefficients w_j, used in G.723 standard, are applied. This metric weights w_j are given by

$w, =1/(f_2 - f_1),$

$w_j -1 / \min(f_j - f_{j-1}, f_{j+'} - f_j), j = 2,\ldots,9,$

$w_{10} = 1/(f_{10} - f_9)$.

In one embodiment of the present invention the following parameters of the quantizer of **Figure 7** are chosen. Denote by *N* the number of codewords in the codebook of the first-stage quantizer 710. In one embodiment, one (first) of these codewords is formed from the previous quantized LSF vector as a predicted LSF vector value, while the rest of the *(N -1)* codewords do not depend on the previous LSF vectors (e.g., they are precomputed using LBG approach). Alternate embodiments use more predicated LSF vectors.

Denote by *M* the number of candidates chosen by the first-stage quantizer. The switch unit forwards to first splitting means those error vectors which correspond to the predicated LSF vector (if the predicated LSF vector is selected as one of the candidates), and it forwards to second splitting means the remaining error vectors. Both splitting means split input 10-dimensional vectors into pair of 5-dimensional vectors. Denote by L_1, L_2, L_3 and L_4 the codebook sizes of codebooks used in MSLQ 1, ..., MSLQ 4 units. These codebooks are also found using the LBG technique. The parameters of the MSLQ units may be chosen by such a way that quantization precision is the same as for a full-search quantization. To achieve a better number of bits/quantization accuracy tradeoff, a variable-length encoding of candidate indices and different sizes L_1, ..., L_4 are used. To meet the fixed total number of bits constraint, a larger codebook is used for those candidates for which the

candidate's codeword length is shorter. An example of bit allocation is shown on **Figure 8**.

The simulation results for different bit rates and bit allocations are shown in Table 3 for fixed rate LSF quantizers with bit rate 15...22 b/frame. The quantization accuracy is characterized by the average weighted distortion (AWD). The AWD for FS-1016 standard scalar 34 bits/frame quantizer and 24 bits/frame vector-split ITU G.723 standard quantizer are given for the comparison.

Quantization scheme						Number of bits per LSF vector	Average weighted distance (dB)
Number of candidates N	List size M	Book sizes					
		L_1	L_2	L_3	L_4		
2	2	*128*	*128*	*128*	*128*	15	6.31
3	3	*256*	*128*	128	128	16	5.51
4	4	*256*	*256*	128	128	17	4.87
3	3	*256*	*256*	256	256	18	4.30
5	4	512	512	256	256	19	3.62
4	4	512	512	512	512	20	3.14
8	4	512	512	512	512	21	2.92
16	4	512	512	512	512	22	2.10
FS-1016 Standard						34	5.73
G.723 Standard						24	2.90

TABLE 3

While the invention has been described in terms of several embodiments, those skilled in the art will recognize that the invention is not limited to the embodiments described. The method and apparatus of the invention can be practiced with modification and alteration within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of limiting on the invention.